structure consists of a plurality of parts and comprises what is referred to as a "key index list" in which all indexed paths of a document are stored. When a query is submitted, said paths are compared in turn with the query until a matching entry is found in the key index list. Based on the information that is stored in the key index list in relation to this entry, the positions in a description stream at which the indexed entry is present in coded form can be determined. As a result of using the key index list it is no longer necessary to decode irrelevant data fragments, so that less memory space is required during a query. However, the linear processing of the key index list is time-consuming and the transfer of all the indexed paths is laborious and resource-intensive.

The document Lam S. W. et al., "Representing lexicons by modified trie for fast partial string matching", Character Recognition Technologies, San Jose, 1-2 Feb. 1993, Bellingham, SPIE, pages 229-237, describes a fast lexical search method wherein an input sequence can have both an indefinite length and also several non-specified letters.

The document Wong R. K. et al., "An XML repository for molecular sequence data", Proceedings IEEE International Symposium on Bio Informatics and Biomedical Engineering", pages 35-42, describes a method wherein a large set of data can be efficiently searched with the aid of a "skip tree".

The object of the invention is therefore to create a method for generating a bit stream from an indexing tree wherein the search for indexed data is made easily and efficiently possible.

This object is achieved according to the features of the independent claims. Developments of the invention may be derived from the dependent claims.

The indexing tree from which a bit stream is generated by means of the method according to the invention comprises a multiplicity of hierarchy levels, each hierarchy level being assigned one or more index nodes and the index nodes containing index data which is sorted in the indexing tree according to one or more predetermined criteria. With the method according to the invention, the index data of the index nodes is inserted into the bit stream and for each index node there is also inserted into the bit stream the information indicating at which position in the bit stream the index data of one or more index nodes of the

Claims

1. A method for generating a bit stream from an indexing tree, wherein

(a) the indexing tree comprises a plurality of hierarchy levels and each hierarchy level is assigned one or more index nodes (K1, K2, K3, K4),

(b) the index nodes (K1, K2, K3, K4) contain index data which is sorted in the indexing tree according to one or more predetermined criteria,

(c) the index node (K1) is designated as the parent node and the index nodes (K2, K3, K4) are designated as child nodes, with at least one child node branching off from the parent node (K1) and said child node being located in a lower hierarchy level,

c h a r a c t e r i z e d i n t h a t

- the index data of the index nodes (K1, K2, K3, K4) is inserted into the bit stream, whereby, following insertion of the index data of the parent node (K1), the index data of that child node (K2) which follows first after the parent node in the indexing tree on account of the sorting is inserted without information indicating at which position the index data of said child node (K2) is located in the bit stream;
- information is inserted into the bit stream in each case for the child node (K3, K4) which does not follow first after the parent node (K1), said information indicating at which position in the bit stream the index data of said child node (K3, K4) is located.

2. The method as claimed in claim 1, wherein the indexing tree is a B tree (= Balanced tree).

3. The method as claimed in claim 1 or 2, wherein the index data is sorted lexicographically in the indexing tree.

4. The method as claimed in one of the preceding claims, wherein the index data is inserted into the bit stream according to the depth-first ordering principle.

5. The method as claimed in one of the preceding claims, wherein the index data comprises paths of a document structure tree consisting of at least one root node and a plurality of leaf nodes.

6. The method as claimed in claim 5, wherein the index data contains the value instances of the paths and the positions of the value instances in the document which is represented by the document structure tree.

7. The method as claimed in claim 5 or 6, wherein the index data comprises the number of paths in an index node (K1, K2, K3, K4).

8. The method as claimed in one of the claims 5 to 7, wherein the paths comprise absolute paths which start from the root node and lead to a leaf node.

9. The method as claimed in one of the claims 5 to 8, wherein the paths comprise relative paths, a relative path of a respective index node (K1, K2, K3, K4) being a path relative to a path, previously inserted into the bit stream, of the respective index node (K1, K2, K3, K4) or of an index node (K1, K2, K3, K4) of a hierarchy level above the hierarchy level of the respective index node (K1, K2, K3, K4).

10. The method as claimed in claim 9, wherein the paths inserted into the bit stream are the paths of the index node whose index data is inserted into the bit stream as the first index data of a hierarchy level in a reverse sequence to the sequence in which the index data is arranged in the index node (K1, K2, K3, K4).

11. The method as claimed in one of the claims 5 to 10, wherein the paths comprise description elements of an XML document (XML = Extensible Markup Language).

12. The method as claimed in claim 11, wherein the paths are XPATH paths of the XML document.

13. The method as claimed in one of the preceding claims, wherein the index data is coded in binary by means of a coding method, in particular by means of an MPEG coding method.

14. The method as claimed in claim 13, wherein the coding method is an MPEG7 coding method.

15. A method for coding a data structure, wherein the data elements of the data structure are indexed in an indexing tree, with a bit stream according to one of the preceding claims being generated and the bit stream being part of the coded data stream.

16. A method for decoding a data structure, wherein the method is embodied in such a way that the data structure coded according to claim 15 is decoded on the basis of the index data in the index nodes.

17. A method for coding and decoding a data structure, comprising the method as claimed in claim 15 and the method as claimed in claim 16.

18. A coding device by means of which a method as claimed in claim 15 can be performed.

19. A decoding device by means of which a method as claimed in claim 16 can be performed.

20. A device for coding and decoding a data structure by means of which a method as claimed in claim 17 can be performed.